

СИСТЕМА ОТСЛЕЖИВАНИЯ БРАКА НА ПРОИЗВОДСТВЕ НА ОСНОВЕ МАШИННОГО ЗРЕНИЯ

Орлов К.Е., Нефедов В.Д., Попов В.В.

Томский политехнический университет, Институт кибернетики
keo1@tpu.ru

Введение

В настоящее время на производстве случается достаточное количество браков, таких как: искривление радиуса у подшипников, неравномерная покраска деталей автомобиля, не соответствие заданных параметров какого-либо изделия и т.д.

Целью моей работы являлась попытка отследить браки и по возможности предотвратить их появление на конвейере.

Описание системы

Компоненты для данной системы были напечатаны на 3D-принтере модели Duplicator 4. Модель данной системы представлена на рис. 1



Рисунок 1 - Макет Системы

Система состоит из:

- Напечатанных деталей (рис. 2);
- соединяющих элементов;
- веб-камеры компании Microsoft (рис. 3)
- Цифровых сервоприводов с крутящим моментом от 2 кг×см до 5 кг×см (рис.4)



Рисунок 2. Напечатанные детали



Рисунок 3 - Веб-камера.



Рисунок 4 - Цифровые сервоприводы.

Программная часть системы, а именно графическая обработка информации выполнена на одноплатном компьютере.

В данный момент система работает следующим образом: одноплатный компьютер, с помощью веб-камеры, распознает объект красного цвета и передает команду сервоприводам сделать поворот на N градусов, с целью постоянного слежения за объектом. Простейший алгоритм слежения за объектом служит в качестве наглядного примера работы системы в целом, в результате которого были выявлены недочеты модели и неудачный подбор сервоприводов.

Тестирование системы

Тестирование системы производилось на основе одноплатного компьютера Raspberry Pi.

Было принято решение кодировать на языке программирования Python с использованием двух сервоприводов из 4. Алгоритм работы python-скрипта состоит из 6 пунктов.

1. Преобразование кадра в формат HSV
2. Фильтрация в заданном диапазоне HSV

3. Морфологическое преобразование
4. Размытие
5. Детектирование окружностей
6. Передача управляющих сигналов на сервоприводы

Преобразование в HSV

hsv = cv2.cvtColor(img, cv.CV_BGR2HSV)

Даем изображение с веб-камеры, получаем - конвертированное в HSV изображение (первая картинка).

В HSV проще создать правильную маску для выделения нужного цвета.

Фильтрация по цвету

thresh = cv2.inRange(hsv, h_min, h_max)

Функция *inRange* преобразует цветную картинку в черно-белую маску. В этой маске, все пиксели, попадающие в заданный диапазон - становятся белыми. Прочие - черными. Результат работы *inRange* представлен на второй картинке.

Морфологическое преобразование

st1 = cv2.getStructuringElement(cv2.MORPH_RECT, (21, 21), (10, 10))

st2 = cv2.getStructuringElement(cv2.MORPH_RECT, (11, 11), (5, 5))

thresh = cv2.morphologyEx(thresh, cv2.MORPH_CLOSE, st1)

thresh = cv2.morphologyEx(thresh, cv2.MORPH_OPEN, st2)

Данная процедура нужна для того, чтобы убрать из кадра мелкий мусор и замазать возможные дефекты в выделяемом объекте. Например, морфологическое преобразование позволяет убрать из теннисного шарика прожилку, которая имеет отличный цвет. Либо, как в моем случае, можно убрать надпись и засвеченные участки шарика (третья картинка).

Размытие

thresh = cv2.GaussianBlur(thresh, (5, 5), 2)

Размытие методом Гаусса. Как и предыдущая процедура, размытие необходимо для сглаживания шероховатостей.

Детектирование окружностей

circles = cv2.HoughCircles(thresh, cv.CV_HOUGH_GRADIENT, 2, mcr, np.array([]), cet1, cet2, mcs, xcs)

Собственно, само детектирование. Процедура *HoughCircles* находит на изображении все окружности, используя при этом преобразование Хафа. Важными параметрами здесь являются:

- *mcr* - минимальное расстояние между окружностями ($h/4$);

- *cet1* и *cet2* - параметры оператора Кэнни, используемого для построения границ объекта (80 и 50);

- *mcs*, *xcs* - минимальный и максимальный радиус окружностей (5 и 0).

На последнем шаге, на основе координат обнаруженной окружности, рассчитываются углы поворота сервоприводов.

*sctrl.shift(0, (x*1./w)*20-10)*

*sctrl.shift(1, -(y*1./h)*20-10).*

Заключение

В данный момент система находится на стадии отладки. В результате тестирования можно сделать вывод, что данная система работоспособна и в дальнейшем может внедряться в производство с изменением алгоритма работы системы. Безусловно, распознавание цвета не самая трудная задача, но в последствии данная система будет совершенствоваться и становиться сложнее.

Список использованных источников

1. Сделай сам [Электронный ресурс]. – URL: <http://www.instructables.com/id/A-fully-3D-printable-GlaDOS-Robotic-ceiling-arm-la/> (Дата обращения 20.04.2015).

2. Популярная робототехника [Электронный ресурс]. – URL: <http://www.poprobot.ru/home/modulslezeniazaoranzevymzarikom> (Дата обращения 25.05.15).